Instruction-tuned Large Language Models for Machine Translation in the Medical Domain

Miguel Rios
Centre for Translation Studies, University of Vienna
miguel.angel.rios.gaona@univie.ac.at

Abstract

Large Language Models (LLMs) have shown promising results in machine translation, particularly for high-resource settings. However, in specialised domains such as medicine, their translation quality underperforms compared to standard Neural Machine Translation models, particularly regarding terminology consistency. In this study, we investigate the impact of instruction tuning for enhancing LLM performance in machine translation for the medical domain. We compare baseline LLMs with instruction-tuned models, and explore the impact of incorporating specialised medical terminology into instruction-formatted fine-tuning datasets. Our results show that instruction tuning significantly improves LLM performance according to automatic metrics. Furthermore, error analysis based on automatic annotation shows a substantial reduction in translation errors in the instruction-tuned models compared to the baselines.

1 Introduction

Current state-of-the-art Large Language Models have shown promising results in machine translation for high-resource language pairs and domains (Bawden and Yvon, 2023). However, in low-resource domains (e.g. medical) LLMs have shown lower performance compared to standard neural machine translation (NMT) models (Bawden and Yvon, 2023; Pourkamali and Sharifi, 2024). The accuracy and consistency in the machine translation of terminology, syntax, and document structure is crucial for users, researchers, and translators who post-edit machine translated documents in high-risk domains (Almahasees et al., 2021; Pang et al., 2024). Moreover, the introduction of in-domain translation constraints during generation into neural models is currently an open problem (Saunders et al., 2019; Alves et al., 2023; Hauhio and Friberg, 2024).

LLMs are trained to perform different Natural Language Processing (NLP) tasks such as summarisation, question answering, and translation, where users interact with the models via instructions (e.g. chat interface) (Touvron et al., 2023; OpenAI et al., 2024; Dubey et al., 2024). Instruction-tuning is a technique that leverages datasets from different NLP tasks, structured as prompts, for fine-tuning LLMs to enhance generalisation across novel tasks and domains (Chung et al., 2022). For example, in machine translation (MT), translating a segment from the European Medicines Agency corpus with specialised terminology the prompt can be framed as: "*Glossary: medicine -> medicamento.*
Translate the source text from English to Spanish following the provided translation glossaries.
English: The medicine was effective in patients with all three types of homocystinuria.
Spanish: ".

Moreover, Alves et al. (2024) instruction-tuned Llama-2 (Touvron et al., 2023) to perform translation related tasks, such as segment and document level translation, post-editing, terminology aware translation, and error annotation. The controlled generation of MT output with the correct terminology, segment length, or syntax can be framed as an instruction-tuning task for LLMs. Thus, improving the workflow of translation during post-editing with an instruction-following (i.e. chat) interface for an LLM tuned on a specific domain.

We seek to answer the following research question: Does instruction-tuning based on terminology rules improve translation quality on LLMs? In this paper, we show results for adding specialised medical dictionaries into fine-tuning for LLMs. In particular, we follow the methodology from (Alves et al., 2024) by incorporating terminology information into the instruction-tuning datasets. Unlike (Alves et al., 2024), our approach relies on openly

Proceedings of Machine Translation Summit XX Volume 1, pages 162–172, June 23-27, 2025

available medical dictionaries and employs simple heuristics to construct instruction-tuning datasets. An instruction-based interface could facilitate the interaction between professional translators and LLMs, and enables model customisation via the integration with user-defined terminology dictionaries.

Our contributions are as follows: We use parameter-efficient fine-tuning (PEFT) and quantisation of large language models (LLMs) for in-domain translation. We leverage medical dictionary term pairs with parallel data to construct prompts that guide LLMs in translating specific terminology.

We evaluate FLAN-T5 (Chung et al., 2022), Llama-3 (Dubey et al., 2024), and Tower (Alves et al., 2024) for English-Spanish, English-German, and English-Romanian language pairs in a split of a medical domain dataset.

The instruction-tuned models outperformed the baseline models with the automatic metrics BLEU (Papineni et al., 2002), chrF (Popović, 2015), and COMET (Rei et al., 2020). Moreover, instruction-tuning improves the overall accuracy of the terminology. Finally, we evaluate the two best models with automatic error annotation (Guerreiro et al., 2024), and quality estimation (Rei et al., 2023).

2 Background and Related Work

Auto-regressive language models predict the next token in a sequence given a prefix context (Jelinek, 1998; Bengio et al., 2000), where LLMs are pre-trained with large amounts of texts followed by fine-tuning on different downstream tasks (OpenAI et al., 2024). In addition, Chung et al. (2022) propose to fine-tune LLMs with a mixture of several NLP datasets into an instruction format to improve: generalisation to unseen tasks, and generation given instruction prompts. For a machine translation task, the LLM is conditioned on a user defined prompt that consists of a translation instruction along with the source text to be translated (Pang et al., 2024). During testing, zero-shot prompting involves querying an LLM with a test input that was not present in the training data. For example, MT instruction includes a prompt asking to translate from a source language to a target language, and the corresponding source text. However, few-shot prompting provides a few examples of the translation task along with the test input to guide the LLM generation. In MT, few-shot examples consist of parallel source and human-translated sentences.

Supervised fine-tuning (SFT) is one of the most popular techniques for domain adaptation in LLMs, where models continue their training with a sample of in-domain data (Alves et al., 2023; Eschbach-Dymanus et al., 2024). However, SFT for LLMs requires large amounts of computational resources, given that during training models update billions of parameters. The goal of Parameter-efficient fine-tuning (PEFT) is to update (i.e. tune) a minimal set of parameters to achieve a similar performance compared to full SFT on downstream tasks. Hu et al. (2021) propose low-Rank adaptation (LoRA) that freezes all the pre-trained model parameters and adds adapter trainable low-rank decomposition matrices of parameters into each layer of the model.

Moreover, Dettmers et al. (2023) propose that during fine-tuning to quantise the parameters of the pre-trained model into fewer bits (e.g. 4-bit) and keep the LoRA adapters with standard precision, thus reducing the memory usage. PEFT and quantisation with QLoRA enables academic translation practitioners to fine-tune LLMs with limited computing resources. Llama versions 2 and 3 (Touvron et al., 2023; Dubey et al., 2024) are open-source LLMs with different parameter scales, which are instruction-tuned for multiple Natural Language Processing tasks. Moreover, Llama has become the base model for the MT related work (Alves et al., 2023; Pang et al., 2024; Eschbach-Dymanus et al., 2024).

Zhang et al. (2023) compared 15 baseline LLMs and fine-tuned with QLoRA on different MT tasks (e.g. segment and document level translation) for the French-English language pair. Llama-2 outperformed other LLMs, fine-tuning improves performance on models that struggle on a few-shot setup, and QLoRA is potentially superior to full SFT in terms of efficiency. Alves et al. (2023) compared instruction tuning with LoRA to few-shot prompting using Llama-2 in various language pairs. Fine-tuning outperforms the few-shot learning, is comparable to full SFT, requires few training data, and tackles over generation. However, LLMs struggle with translation directions out of English (en-xx). Alves et al. (2024) proposed Tower with a focus on translation related tasks, for example, document level translation, post-editing, and terminology-aware prompts. Tower is based on the continued training of Llama-2 with parallel translation data, and is followed by instruction-tuning for

the MT tasks.

Zheng et al. (2024) proposed to fine-tune LLMs based on prompts, and compared it to LoRA for domain adaptation in IT for Chinese-English and English-Chinese MT. Moreover, Zheng et al. (2024) incorporate IT terminology by few-shot prompting and chain-of-thought. The template used for the proposed prompt-tuning model has a substantial impact on performance, and the introduction of terminology with simple prompt rephrasing outperforms chain-of-thought. Eschbach-Dymanus et al. (2024) studied domain adaptation for business IT with LLMs. They compared full SFT, LoRA, different prompting techniques, and standard NMT. Finally, Eschbach-Dymanus et al. (2024) defined guidelines for domain adaptation with LLMs. Moslem et al. (2023) evaluate LLMs for translation on specialised domains (e.g. medical COVID-19), and incorporate terms from glossaries into their prompts to tackle issues with no retrieved matches in few-shot learning. Jerpelea et al. (2025) developed a parallel dataset for the low-resource languages Romanian, and Aromanian, they instruction-tuned Llama-3 for Romanian, and compared multilingual NMT, GPT, Llama-3, and Tower for translation.

We followed (Alves et al., 2023, 2024) for our experimental design. Unlike previous work on LLM for MT, our approach focuses on the medical domain, relies on openly available medical dictionaries, employs simple heuristics to construct the instruction-tuning datasets, and uses efficient tuning techniques. In particular, we evaluate the impact of instruction tuning for improving terminology translation in LLMs.

3 Experimental Setup

3.1 Data

We use the corpus of the European Medicines Agency (EMEA) (elr) for the English-Spanish (en-es), English-German (en-de), and English-Romanian (en-ro) language pairs. The EMEA corpus contains multilingual PDF documents from the European Medicines Agency, automatically converted to text and aligned at the segment level. We randomly split the EMEA corpus into 20K segments for each language pair. These subsets were then merged into a single tuning dataset of 60K segments. Furthermore, we created separate validation and test sets, each containing 500 segments per language pair.

3.2 Terminology Annotation

The Interactive Terminology for Europe (IATE)[1] is a terminology management system from EU institutions that covers different domains (e.g. economics, law, health). For our source and target language pairs, we downloaded the IATE database in the *health* domain (*id 2841*). We only used terms with quality 3 (reliable) and 4 (very reliable) stars (human annotated quality scores), resulting in 38,898 terms for en-es, 49,828 terms for en-de, and 9,551 terms for en-ro.

We incorporate medical terms as translation instructions by identifying term pairs within each aligned segment. For every aligned segment, we retrieve candidate terms using strict matching, which requires the presence of a candidate pair in both the source and target segments. If one or more candidate pairs are identified, we include them in the instruction template within the prompt. For example, an instruction-tuning input in en-es "*spectrum of activity -> espectro de actividad, amoxicillin -> amoxicilina, and activity -> actividad*" are term pairs identified in the parallel segment:

```
Glossaries:
"spectrum of activity" -> "espectro de
    actividad"
"amoxicillin" -> "amoxicilina"
"activity" -> "actividad"
Translate the source text from English
    to Spanish following the provided
    translation glossaries.
English: Amoxicillin is susceptible to
    degradation by beta-lactamases
    produced by resistant bacteria and
    therefore the spectrum of activity
    of amoxicillin alone does not
    include organisms which produce
    these enzymes.
Spanish: La amoxicilina es sensible a
    la degradación por las beta-
    lactamasas producidas por bacterias
     resistentes y por tanto el
    espectro de actividad de la
    amoxicilina sola no incluye
    microorganismos productores de
    estas enzimas.
```

When no candidates are identified within a segment, the instruction consists only of the translation task prompt. For example, an instruction-tuning input in en-es:

```
Translate the source text from English
    to Spanish.
English: Do not use Cymevene if you are
    breast-feeding.
Spanish: No use Cymevene si está en
    periodo de lactancia.
```

[1] https://iate.europa.eu/download-iate

The prompt templates for the baseline models are defined in Section B, and we perform zero-shot prompting to generate translations for en-es, en-de, and en-ro. For example, a test input in en-es:

```
Glossary:
"insulin" -> "insulina"
Translate the source text from English
    to Spanish following the provided
    translation glossaries.
English: Within-subject variability of
    the time action profile of Levemir
    and NPH insulin Pharmacodynamic
    Endpoint
Spanish:
```

3.3 Models

We use the HuggingFace transformers framework for the baseline LLMs (Wolf et al., 2020), and PEFT (Mangrulkar et al., 2022) for the instruction-tuning with QLoRA. Our baseline LLMs are as follows: FLAN-T5-large[2] (783M parameters), an encoder-decoder model; Llama-3-8B[3], an instruction-tuned LLM for NLP tasks; and Tower-7B[4], an instruction-tuned LLM for MT tasks. We evaluated two distinct instruction-tuned model architectures: encoder-decoder model based on FLAN-T5, and auto-regressive LLMs based on Llama.

We use QLoRA with a 4-bit quantisation to fine-tune each baseline model for one epoch on the tuning dataset (60K segments). The values for QLoRA and tuning hyper-parameters for each model are defined in Section A, for FLAN-T5 7, Llama-3-8B 8, and Tower-7B 9. Finally, for generation, we use zero-shot prompting and stochastic decoding with top-p sampling $p = 0.9$. We release our scripts and data on GitHub at: https://github.com/HAITrans-lab/instruction-tuned-medical-LLM

4 Results

We show results with automatic metrics and terminology accuracy for FLAN-T5, Llama-3 and Tower for the en-es, en-de and en-ro language pairs. Moreover, we show automatic error annotation, and quality estimation scores for the best performing models.

4.1 Automatic Metrics

We evaluated all models with BLEU, chrF, and COMET in the test split. Table 1 shows the comparison between the baselines and the instruction-tuned models with QLoRA. The BLEU, chrF, and COMET scores for the instruction-tuned models are statistically significant ($p < 0.05$) for all models.

To prevent over-generation and improve the performance of Llama-3, we post-processed the output by cutting it at the first appearance of the end-of-sequence token "<|*eot_id*|>". As noted by Zhang et al. (2023), Llama models repeat the translation output or produce *assistant* suggestions to improve the prompts along with the translation.

In Table 1, Tower and the QLoRA Tower outperform the other models with the automatic metrics for en-es, and en-de. However, Romanian (en-ro) is not present in the original Tower fine-tuning for MT. Tower is based on LLaMA-2 which is not focused on multilingual data, in contrast to Llama-3. Moreover, QLoRA tuning produced improvements for all models.

As shown in Table 1, Tower and QLoRA Tower achieved the highest automatic metric scores for en-es, and en-de. However, the original Tower model was not fine-tuned for en-ro MT. Furthermore, Tower is based on LLaMA-2, which is less focused on multilingual data compared to Llama-3. Nonetheless, the QLoRA models consistently improved performance across all models. The bold numbers are the best automatic scores across all models for a given language pair.

Terminology Accuracy We compute the accuracy of the terminology in the MT output compared to the reference translations. To compute accuracy, the exact term must be present in both the MT segment and the database to be correct. Table 2 shows the accuracy scores for the terminology. Instruction-tuning improves the accuracy of terms across models, where Flan-T5 followed by Tower achieve the highest terminology accuracy performance. We observed that the LLM produced translations with increased terminology accuracy for the high-resource language pairs, en-de and en-es.

4.2 Automatic Error Annotation

Automatic metrics are not designed to identify specific translation errors in MT outputs, for example, errors in terminology. Multidimensional Quality Metrics (MQM) (Lommel et al., 2014) are based on manually classifying and annotating errors using predefined categories. The MQM error typology

[2] google/flan-t5-large
[3] meta-llama/Meta-Llama-3.1-8B-Instruct
[4] Unbabel/TowerInstruct-7B-v0.2

Model	en-es↑			en-de↑			en-ro↑		
	BLEU	chrF	COMET	BLEU	chrF	COMET	BLEU	chrF	COMET
FLAN-T5	28.51	57.11	0.73	14.76	43.86	0.63	17.34	45.00	0.64
QLoRA FLAN-T5	36.43	63.40	0.78	25.45	54.93	0.72	28.65	57.44	0.77
Llama-3-8B	34.07	63.02	0.79	25.44	58.08	0.78	24.99	53.17	0.76
QLoRA Llama-3-8B	45.07	67.74	0.85	36.30	62.21	0.84	**35.97**	**61.19**	**0.85**
Tower-7B	42.27	66.31	0.86	34.80	62.45	0.85	18.20	44.86	0.69
QLoRA Tower-7B	**48.88**	**70.36**	**0.87**	**42.11**	**67.62**	**0.87**	23.93	50.57	0.78

Table 1: Comparing the baseline and QLoRA fine-tuned LLMs with **automatic metrics** for the en-es, en-de, and en-ro language pairs.

Model	en-es↑	en-de↑	en-ro↑
FLAN-T5	0.72	0.45	0.38
QLoRA FLAN-T5	0.90	**0.91**	**0.90**
Llama-3-8B	0.59	0.53	0.44
QLoRA Llama-3-8B	0.69	0.68	0.51
Tower-7B	0.88	0.79	0.58
QLoRA Tower-7B	**0.91**	0.86	0.68

Table 2: Comparing the baseline and QLoRA fine-tuned LLMs with **terminology accuracy** for the en-es, en-de, and en-ro language pairs.

covers high-level error categories (e.g. Accuracy, linguistic conventions, style, etc.), where each category can be further expanded into fine-grained categories (e.g. Accuracy into Mistranslation, addition, untranslated, etc.). Expert translators identify an error in the MT output, label it with a category from the typology, and also assign a severity score to it. The severity weights defined in (Freitag et al., 2021) are: minor $\times$ 1 (MIN), major $\times$ 5 (MAJOR), and critical $\times$ 10 (CRIT). The MQM score is defined as follows:

$$\text{MQM} = 100 \cdot \left(1 - \frac{10 \cdot \text{critical} + 5 \cdot \text{major} + \text{minor}}{\text{tokens}}\right), \quad (1)$$

We use XCOMET (Guerreiro et al., 2024) to produce automatic MQM annotations. XCOMET only annotates the error spans in the MT output with severities [5], and the corresponding prediction confidence for each span. The automatic error annotation with XCOMET is based on an LLM that required a larger GPU than our available resources for execution. We run the XCOMET evaluations on CPU where the process is slow, thus we only evaluate the best two models based on the automatic metrics, Llama-3 and Tower.

We show the number of errors ↓ in Table 3 and the MQM scores ↑ in Table 4 for each system. The MQM score summarises the individual errors into a weighted score based on severity (Equation 1). Table 3 shows the number or errors by severity for each model. The total number of errors for the instruction-tuned Llama-3 is: 1914 (en-es), 2910 (en-de), and 1764 (en-ro). The instruction-tuned Tower is: 745 (en-es), 1059 (en-de), and 1632 (en-ro). Instruction-tuned Tower shows fewer critical errors compared to Llama for the three language pairs (en-es, en-de, and en-ro).

Table 4 presents the MQM scores, which show a reduction in critical, major, and minor errors after the instruction tuning phase. In these results, Tower outperforms Llama.

Automatic Error Annotation Analysis We conducted a preliminary error analysis of the automatic error annotation for en-es to assess the quality of XCOMET to label translation errors. A native Spanish speaker with English proficiency served as the annotator. The limited number of examples analysed from the en-es automatic error annotation is because of the lack of a professional medical translator during the preliminary analysis. We show annotation examples between the baseline and instruction-tuned models for Llama-3 and Tower.

Table 5 presents examples of automatic error annotations generated by XCOMET for Llama, QLoRA Llama, and Tower. For Llama, XCOMET identified a critical error with a confidence score of 0.52. Similarly, a critical error in the instruction-tuned Llama was annotated with a confidence of 0.40. While XCOMET produced incomplete annotations, potentially because of over-generation by Llama, it successfully identified code-switched words, such as "*assistant*".

In Tower, XCOMET annotated a major error, "*reconstitución,*" with a confidence of 0.50. For

[5] Unbabel/XCOMET-XL

Model	en-es↓			en-de↓			en-ro↓		
	MIN	MAJ	CRIT	MIN	MAJ	CRIT	MIN	MAJ	CRIT
Llama-3-8B	145	1277	1240	1693	719	938	95	983	1301
QLoRA Llama-3-8B	359	1105	450	2160	295	455	225	844	695
Tower-7B	592	241	15	1266	50	**25**	253	844	695
QLoRA Tower-7B	583	149	**13**	1007	26	26	503	868	**261**

Table 3: Comparing the baseline and QLoRA fine-tuned LLMs with the number of **errors** with the following categories: minor (MIN), major (MAJ), and critical (CRIT).

Model	en-es↑	en-de↑	en-ro↑
Llama-3-8B	35.98	41.29	27.76
QLoRA Llama-3-8B	58.83	59.45	**45.66**
Tower-7B	82.35	80.70	20.11
QLoRA Tower-7B	**86.63**	**84.69**	36.96

Table 4: Comparing the baseline and QLoRA fine-tuned LLMs with **MQM scores** for the en-es, en-de, and en-ro language pairs.

the instruction-tuned Tower, a minor error, "*reconstitu,*" was annotated with a confidence of 0.42. Notably, "*reconstitución*" is the correct term in the MT output with low prediction confidence. A potential solution involves filtering annotations based on a predefined confidence threshold, keeping only high-confidence predictions.

4.3 Quality Estimation

Quality estimation (QE) models predict a quality score for the MT output without using reference translations. QE evaluation can be useful for cases of low-resource language pairs and practical applications, given the lack of reference translations. We use COMETKiwi (Rei et al., 2023) for QE evaluation [6]. COMETKiwi is based on COMET features to train a QE prediction model. The QE model is trained with an annotated multilingual source and corresponding MT outputs to predict quality based on direct assessment (i.e. ranking) or MQM scores.

Table 6 shows the comparison of QE scores for Llama-3 and Tower. The instruction-tuned Tower shows higher QE scores compared to Llama in all language pairs. The QE scores show a similar order in model quality compared to the output of automatic metrics without the need for reference translations.

[6] Unbabel/wmt22-cometkiwi-da

4.4 Discussion and Limitations

Instruction-tuning improves the overall accuracy of terminology and translation quality (e.g. automatic metrics). Instruction-tuned FLAN-T5 (encoder-decoder) has the highest terminology accuracy, but its improvements in translation quality are lower compared to the LLMs. A possible explanation is the difference in parameter size compared to the LLMs, and pre-trained data available for the LLMs. However, to achieve a more accurate evaluation, it is recommended to perform a manual error annotation with professional medical translators.

Both the baseline and instruction-tuned models generate terms defined by our prompts. However, fine-tuning substantially improves accuracy for FLAN-T5, Tower, and Llama-3. Furthermore, Tower includes terminology translation across diverse domains as a component of its tuning tasks.

Llama-3 presents over-generation, producing an excessive amount of tokens with assistant suggestions. For example, in en-es in the test set, the baseline model generates 29,569 tokens, which is reduced to 25,225 tokens after fine-tuning. Examples of this over-generation in Llama-3 include assistant-specific text alongside the expected machine translation output, such as: "*..{source segment} assistant Here is the corrected translation: {MT target segment}...*". However, the instruction-tuned LlaMA-3 also over-generates: *"..I corrected the translation using the provided glossary.assistant Using the glossary..."*, or it continues repeating the MT output. A possible solution is to use a prompt that constrains the model to produce only the target segment. With our current prompt, both Llama-3 models require extra post-processing to extract the MT and avoid biases on the automatic metrics and automatic error annotation. On the other hand, Tower generates 11,034 tokens compared to 10,906 tokens for the instruction-tuned. The MT tasks tuning on Tower improves translation accuracy and avoids over-generation.

Terms	Source, reference, and MT	Annotation
active substance ->principio activo, system ->sistema, fentanyl ->fentanilo	**src**: Ionsys transdermal system delivers the active substance, fentanyl, through the skin.	**critical error**: assistant Here is the translation: Ionsys sistema transdérmico proporciona
	ref: El sistema de liberación transdérmica Ionsys administra el principio activo, el fentanilo, a través de la piel.	
	Llama: Ionsys sistema transdérmico proporciona la sustancia activa, fentanilo, a través de la piel.**assistant Here is the translation: Ionsys sistema transdérmico proporciona** el principio activo, fentanilo, a través de la piel. I corrected the translation using the provided glossary.assistant Using the glossary "system" ->"sistema", I should have translated it to	
active substance ->principio activo, system ->sistema, fentanyl ->fentanilo	**src**: Ionsys transdermal system delivers the active substance, fentanyl, through the skin.	**critical error**: assistant El sistema transdérmico Ionsys libera el
	ref: El sistema de liberación transdérmica Ionsys administra el principio activo, el fentanilo, a través de la piel.	
	QLoRA Llama: El sistema transdérmico Ionsys libera el principio activo, el fentanilo, a través de la piel. **assistant El sistema transdérmico Ionsys libera el** principio activo, el fentanilo, a través de la piel.	
reconstitution ->reconstitución	**src**: Write the date of reconstitution and expiry on the label (expiry is 1 month after reconstitution)	**major error**: reconstitución
	ref: Escriba la fecha de reconstitución y la de caducidad en la etiqueta (la caducidad es 1 mes después de la reconstitución)	
	Tower: Escriba la fecha de reconstitución y el de caducidad en la etiqueta (el de caducidad es 1 mes después de la **reconstitución**).	

Table 5: Examples of **automatic error annotation** for en-es using XCOMET.

Model	en-es↑	en-de↑	en-ro↑
Llama-3-8B	0.513	0.507	0.484
QLoRA Llama-3-8B	0.657	0.619	0.595
Tower-7B	0.840	0.806	0.647
QLoRA Tower-7B	**0.850**	**0.825**	**0.754**

Table 6: Comparing the baseline and QLoRA fine-tuned LLMs with **QE** for the en-es, en-de, and en-ro language pairs.

However, given common limitations on academic computational resources (one GPU) we use small size LLMs (8B) with quantisation, PEFT for tuning our models, and a small split of the EMEA corpus. A limitation of quantisation is the use of pre-trained models with lower precision models that may hurt overall performance. However, SFT in LLMs can be achieved with significantly less data than training from scratch and other domain adaptation approaches (Zhu et al., 2024). The total size of the EMEA corpus is approximately 1M segments.

Automatic error annotation and QE scores offer a detailed evaluation of our language pairs and domain. However, XCOMET shows inaccuracies in terminology annotation, particularly with low-confidence predictions. Furthermore, to validate the reliability of automatic error annotation within the medical domain, a comprehensive analysis involving professional translators is essential. Additionally, XCOMET requires substantial GPU resources.

We use accuracy to evaluate the terms generated in the MT output. The limitation of accuracy is that context is not taken into account (Corral and Saralegi, 2024), for example, translation quality is lowered with high term accuracy in FLAN-T5. A limitation of building our terminology prompt dataset using only exact matches is the potential to miss terms that are expressed differently depending on the context. Furthermore, the coverage of terms and domains within IATE represents a limitation of terminology databases. For example, in the parallel (en-es) segment: *"Posology for MDS/MPD The recommended dose of imatinib is 400 mg/day for adult patients with MDS/MPD."* and *"Posología para SMD/SMP La dosis recomendada de imatinib para pacientes adultos con SMD/SMP es de 400 mg/día"* from IATE the exact term match is "*dose -> dosis*". However, IATE does not contain "*MDS/MPD -> SMD/SMP*" that means "*Myelodysplastic/Myeloproliferative Neoplasm*"[7]. A possible solution is to combine translation terminology databases with medical ontologies, for example, Medical Subject Headings (MeSH)[8], and the Unified Medical Language System (UMLS)[9] that has multilingual features.

[7] https://www.cancer.gov/types/myeloproliferative/hp/mds-mpd-treatment-pdq

[8] https://www.nlm.nih.gov/mesh/meshhome.html

[9] https://www.nlm.nih.gov/research/umls/index.html

5 Conclusions and Future Work

In this study, we show a comparison between baseline LLMs and QLoRA instruction-tuned models in the medical domain for en-es, en-de, and en-ro. We introduce medical terminology from IATE into an instruction-formatted dataset for controlled generation in LLMs. Instruction-tuned models significantly outperform the baseline across automatic evaluation metrics. Furthermore, these models show improved accuracy in terminology translation compared to the baseline.

In particular, the instruction-tuned Tower model presents superior translation quality according to different evaluation methods (automatic metrics, MQM annotation, and QE). Additionally, Tower requires fewer computational resources and less post-processing compared to LLaMA-3.

A limitation of our current evaluation is the reliance on automatic metrics and the limited quality of automatic error annotation. For future work, we will evaluate the baselines with few-shot instead of zero-shot. We will define different prompts for Llama-3 to avoid over-generation. We will perform an evaluation on a balanced test split in terms of the number and type of present terms with respect to the training data. Finally, we will perform a manual error annotation, as automatic metrics may not test for correct terminology generation on the MT output (Haque et al., 2019; Gaona et al., 2023).

Sustainability Statement For the experiments we use a Tesla T4 GPU (16GB) from Azure with an approximate SFT time of 20 hours per model. Instruction-tuning with PEFT tackles issues for scare computational resources (GPUs) for short training time (e.g. one epoch) and small tuning data (60K segments). Moreover, we performed automatic error annotation on the CPU instead of GPU given our academic computational limitations.

From MachineLearning Impact calculator presented in (Lacoste et al., 2019): West-Europe Azure has a carbon efficiency of 0.57 $kgCO_2eq/kWh$. A cumulative of 100 hours of computation was performed on hardware of type T4 (TDP of 70W). Total emissions are estimated to be 3.99 $kgCO_2eq$ of which 100 percent were directly offset by the cloud provider.

Acknowledgments

This work was supported by the ZID of the University of Vienna with Azure cloud credits.

References

ELRC3.0 Multilingual corpus made out of PDF documents from the European Medicines Agency (EMEA), (February 2020) ELRC-SHARE.

Zakaryia Almahasees, Samah Meqdadi, and Yousef Albudairi. 2021. Evaluation of google translate in rendering english covid-19 texts into arabic. *Journal of Language and Linguistic Studies*, 17(4):2065–2080.

Duarte M. Alves, Nuno M. Guerreiro, João Alves, José Pombal, Ricardo Rei, José G. C. de Souza, Pierre Colombo, and André F. T. Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. *Preprint*, arXiv:2310.13448.

Duarte M. Alves, José P. Pombal, Nuno M. Guerreiro, Pedro H. Martins, Joao Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *ArXiv*, abs/2402.17733.

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170, Tampere, Finland. European Association for Machine Translation.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *NeurIPS*, volume 13. MIT Press.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Ander Corral and Xabier Saralegi. 2024. Morphology aware source term masking for terminology-constrained NMT. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1676–1688, St. Julian's, Malta. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Abhimanyu Dubey et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Johannes Eschbach-Dymanus, Frank Essenberger, Bianka Buschbeck, and Miriam Exel. 2024. Exploring the effectiveness of llm domain adaptation for business it machine translation. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation*.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Miguel Angel Rios Gaona, Raluca-Maria Chereji, Alina Secara, and Dragos Ciobanu. 2023. Quality analysis of multilingual neural machine translation systems and reference test translations for the English-Romanian language pair in the medical domain. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 355–364, Tampere, Finland. European Association for Machine Translation.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Rejwanul Haque, Md Hasanuzzaman, and Andy Way. 2019. Investigating terminology translation in statistical and neural machine translation: A case study on English-to-Hindi and Hindi-to-English. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 437–446, Varna, Bulgaria. INCOMA Ltd.

Iikka Hauhio and Théo Kalevi Max Friberg. 2024. Mitra: Improving terminologically constrained translation quality with backtranslations and flag diacritics. In *Proceedings of the 25th Annual Conference of The European Association for Machine Translation*, Switzerland. European Association for Machine Translation. Annual Conference of The European Association for Machine Translation, EAMT ; Conference date: 24-06-2024 Through 27-06-2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Frederick Jelinek. 1998. *Statistical Methods for Speech Recognition (Language, Speech, and Communication)*. The MIT Press.

Alexandru-Iulius Jerpelea, Alina Radoi, and Sergiu Nisioi. 2025. Dialectal and low resource machine translation for Aromanian. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7209–7228, Abu Dhabi, UAE. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional Quality Metrics (MQM): A Framework for Declaring and Describing Translation Quality Metrics. *Revista Tradumàtica: tecnologies de la traducció*, (12):455–463. Publisher: Universitat Autònoma de Barcelona.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland. European Association for Machine Translation.

OpenAI et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jianhui Pang, Fanghua Ye, Longyue Wang, Dian Yu, Derek F. Wong, Shuming Shi, and Zhaopeng Tu. 2024. Salute the classic: Revisiting challenges of machine translation in the age of large language models. *Preprint*, arXiv:2401.08350.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Nooshin Pourkamali and Shler Ebrahim Sharifi. 2024. Machine translation with large language models: Prompt engineering for persian, english, and russian directions. *Preprint*, arXiv:2401.08429.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2019. Domain adaptive inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 222–228, Florence, Italy. Association for Computational Linguistics.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with QLoRA. In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, Singapore. Association for Computational Linguistics.

Jiawei Zheng, Hanghai Hong, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. 2024. Fine-tuning large language models for domain-specific machine translation. *Preprint*, arXiv:2402.15061.

Dawei Zhu, Pinzhen Chen, Miaoran Zhang, Barry Haddow, Xiaoyu Shen, and Dietrich Klakow. 2024. Fine-tuning large language models to translate: Will a touch of noisy data in misaligned languages suffice? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 388–409, Miami, Florida, USA. Association for Computational Linguistics.

A Hyper-parameters

The hyper-parameter values tables for FLAN-T5, Llama-3-8B, and Tower-7B are as follows:

Hyper-parameter	Value
r	8
α	32
Dropout	0.1
Target modules	q, v
Max source length	512
Max target length	512
Batch size	6
Learning rate	$2e-4$
Warm-up steps	0.03
Scheduler type	linear

Table 7: FLAN-T5 seq2seq hyper-parameter values. The upper section contains the QLoRA hyper-parameters, and the lower section contains the overall fine-tuning.

Hyper-parameter	Value
r	64
α	128
Dropout	0.05
Target modules	q_proj, v_proj
Max sequence length	512
Batch size	2
Gradient accumulation	4
Learning rate	$2e-4$
Warm-up steps	0.03
Scheduler type	cosine

Table 8: Llama-3-8B hyper-parameter values. The upper section contains the QLoRA hyper-parameters, and the lower section contains the overall fine-tuning.

Hyper-parameter	Value
r	64
α	16
Dropout	0.1
Target modules	q_proj, k_proj, v_proj, o_proj
Max sequence length	512
Batch size	2
Gradient accumulation	2
Learning rate	$2e-5$
Warm-up steps	0.03
Scheduler type	cosine

Table 9: Tower-7B hyper-parameter values. The upper section contains the QLoRA hyper-parameters, and the lower section contains the overall fine-tuning.

B Instruction Templates

Instruction templates for FLAN-T5, Llama-3 and Tower. The source_term is the source entry from IATE, the target_term is the target entry from IATE, source_language is the source language (i.e. English), target_id is the target language (i.e. Spanish, German, and Romanian), and glossary_type is Glossary with one candidate term pair or Glossaries with several candidate terms.

FLAN-T5 instruction template for a segment with an identified pair of candidate terms. The

prompt is the input for the encoder and the target segment is the input for the decoder:

```
{glossary_type}:
"{source_term}" -> "{target_term}"
...
Translate the source text from {
    source_id} to {target_id} following
    the provided translation glossaries.
{source_id}: {source_segment}
```

FLAN-T5 instruction template with a segment without candidate terms. The prompt is the input for the encoder, and the target segment is the input for the decoder:

```
Translate the source text from {
    source_id} to {target_id}.
{source_id}: {source_segment}
```

Llama-3-8B instruction template for a segment with candidate term pairs:

```
<|begin_of_text|><|start_header_id|>
    system<|end_header_id|>
You are a helpful translation assistant
    .<|eot_id|><|start_header_id|>user<|
    end_header_id|>
{glossary_type}:
"{source_term}" -> "{target_term}"
...
Translate the source text from {
    source_id} to {target_id} following
    the provided translation glossaries.
{source_id}: {source_segment}
{target_id}:<|eot_id|>
<|start_header_id|>assistant<|
    end_header_id|>
{target_segment}<|eot_id|>
```

Llama-3-8B instruction template for a segment without candidate term pairs:

```
<|begin_of_text|><|start_header_id|>
    system<|end_header_id|>
You are a helpful translation assistant
    .<|eot_id|><|start_header_id|>user<|
    end_header_id|>
Translate the source text from {
    source_id} to {target_id}.
{source_id}: {source_segment}
{target_id}:<|eot_id|>
<|start_header_id|>assistant<|
    end_header_id|>
{target_segment}<|eot_id|>
```

Tower-7B instruction template for a segment with candidate term pairs:

```
<|im_start|>user
{glossary_type}:
"{source_term}" -> "{target_term}"
...
Translate the source text from {
    source_id} to {target_id} following
    the provided translation glossaries.
{source_id}: {source_segment}
{target_id}:<|im_end|>
<|im_start|>assistant
{target_segment}<|im end|>
```

Tower-7B instruction template for a segment without candidate term pairs:

```
<|im_start|>user
Translate the source text from {
    source_id} to {target_id}.
{source_id}: {source_segment}
{target_id}:<|im_end|>
<|im_start|>assistant
{target_segment}<|im end|>
```